

Amendments to the Claims

1. (Currently amended) A method for joining a plurality of tables T1 through TN, where each of the tables has an associated Star Map, S1 through SN, respectively, and each Star Map includes bitmap entries having locations indexed by ~~the~~ a hash of one or more values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates ~~the presence of a row~~ is present in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry, the method including:

- a) performing a Boolean operation using the bitmap entries of the Star Maps S1 through SN to produce a join Star Map SJ;
- b) using SJ to select rows from the tables T1 through TN; and
- c) joining the resulting rows to produce a join result.

2. (Original) The method of claim 1 where the Boolean operation includes a logical AND.

3. (Original) The method of claim 1 where using SJ to select rows from the tables T1 through TN includes for each set bitmap entry in SJ, searching T1 through TN for all rows having values in their one or more join key columns that hash to the location of the set bitmap entry.

4. (Original) The method of claim 3 where

each bitmap entry in each Star Map has an associated row number and column number;

the row number and the column number for a set bitmap entry in a Star Map can be combined to produce a value that corresponds to the hash value of the one or more join key columns of the Star Map's associated table; and

at least one of the tables T1 through TN is ordered by the hash of the values of its one or more join key columns.

5. (Currently amended) A method for joining a plurality of tables T1 through TN, where each of the tables has an associated Star Map, S1 through SN, respectively, and each Star Map includes bitmap entries having locations indexed by ~~the a~~ hash of one or more values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates ~~the presence of a row~~ is present in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry, the method including:

- a) logically ANDing the Star Maps S1 through SN to produce Star Map SJ;
- b) for each set bitmap entry in SJ, searching T1 through TN for all rows having values in their one or more join key columns that hash to the location of the set bitmap entry; and
- c) joining the resulting rows to produce a join result.

6. (Canceled)

7. (Original) The method of claim 5 where logically ANDing the Star Maps includes setting an entry in SJ having a location defined by a hash value if the entries in all of S1 through SN having locations defined by the hash value are set.

8. (Original) The method of claim 5 where logically ANDing the Star Maps includes

- a1) performing a logical AND of S1 and S2 to produce SJ;
- a2) setting i to 3;
- a3) while i is less than or equal to N, performing a4) through a6);
 - a4) performing a logical AND of Si with SJ to form Stemp;
 - a5) setting SJ equal to Stemp; and
 - a6) incrementing i.

9. (Original) The method of claim 8 where S1 through SN and SJ are the same size and performing a logical AND of Si with SJ to form Stemp includes

ANDing the entry at a location in Si defined by a hash value with the entry at a location in SJ defined by the hash value.

10. (Original) The method of claim 5 where each of T1 through TN are indexed by the hash of the combined entries in its respective one or more join key columns, and searching T1 through TN for rows having values in its one or more columns that hash to the location of the set bitmap entry includes

using the hash value associated with the location of the set bitmap entry as an index to retrieve rows from tables T1 through TN.

11. (Original) The method of claim 5 where one or more of the Star Maps is a table and a first portion of the hash value that indexes the locations of a Star Map defines a row within the Star Map and a second portion of the hash value defines an offset within the defined row.

12. (Original) The method of claim 11 where the first portion is the first half of the hash value and the second portion is the second half of the hash value.

13. (Original) The method of claim 11 where the hash value is 32 bits long, the first portion is the first 16 bits of the hash value, and the second portion is the second 16 bits of the hash value.

14. (Original) The method of claim 5 where
at least one of tables T1 through TN is ordered by the values of its one or more join key columns.

15. (Original) The method of claim 5 where each entry in a Star Map is one bit.

16. (Original) The method of claim 5 where each entry in a Star Map is sixteen bits.

17. (Original) The method of claim 5 where each Star Map entry includes one or more bits and each bit corresponds to the hash of one or more values associated with the one or more join key columns of its associated table.

18. (Currently amended) A computer program, stored on a tangible storage medium, for use in joining a plurality of tables T1 through TN, where each of the tables has an associated Star Map, S1 through SN, respectively, and each Star Map includes bitmap entries having locations indexed by the a hash of one or more values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates the ~~presence of a row~~ is present in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry, the program including executable instructions that cause a computer to

- a) logically AND the Star Maps S1 through SN to produce Star Map SJ;
- b) for each set bitmap entry in SJ, search T1 through TN for all rows having values in their one or more join key columns that hash to the location of the set bitmap entry; and
- c) join the resulting rows to produce a join result.

19. (Canceled)

20. (Original) The computer program of claim 18 where when logically ANDing the Star Maps the computer

sets an entry in SJ having a location defined by a hash value if the entries in all of S1 through SN having locations defined by the hash value are set.

21. (Original) The computer program of claim 18 where when logically ANDing the Star Maps the computer

- a1) performs a logical AND of S1 and S2 to produce SJ;
- a2) sets i to 3;
- a3) while i is less than or equal to N, performs a4) through a6);
 - a4) performs a logical AND of Si with SJ to form Stemp;
 - a5) sets SJ equal to Stemp;
 - a6) increments i.

22. (Original) The computer program of claim 21 where S1 through SN and SJ are the same size and when performing a logical AND of Si with SJ to form Stemp the computer

ANDs the entry at a location in Si defined by a hash value with the entry at a location in SJ defined by the hash value.

23. (Original) The computer program of claim 18 where each of T1 through TN are indexed by the hash of the combined entries in its respective one or more join key columns, and when searching T1 through TN for rows having values in its one or more columns that hash to the location of the set bitmap entry the computer

uses the hash value associated with the location of the set bitmap entry as an index to retrieve rows from tables T1 through TN.

24. (Original) The computer program of claim 18 where one or more of the Star Maps is a table and a first portion of the hash value that indexes the locations of a Star Map defines a row within the Star Map and a second portion of the hash value defines an offset within the defined row.

25. (Original) The computer program of claim 24 where the first portion is the first half of the hash value and the second portion is the second half of the hash value.

26. (Original) The computer program of claim 24 where the hash value is 32 bits long, the first portion is the first 16 bits of the hash value, and the second portion is the second 16 bits of the hash value.

27. (Original) The computer program of claim 18 where
at least one of tables T1 through TN is ordered by the values of its one or more join key columns.

28. (Original) The computer program of claim 18 where each entry in a Star Map is one bit.

29. (Original) The computer program of claim 18 where each entry in a Star Map is sixteen bits.

30. (Original) The computer program of claim 18 where each Star Map entry includes one or more bits and each bit corresponds to the hash of one or more values associated with the one or more join key columns of its associated table.

31. (Currently amended) A database system for accessing a database, the database system including

- a massively parallel processing system including

- one or more nodes;

- a plurality of CPUs, each of the one or more nodes providing access to one or more CPUs;

- a plurality of virtual processes each of the one or more CPUs providing access to one or more processes;

- each process configured to manage data stored in one of a plurality of data-storage facilities;

- a plurality of tables T1 through TN, each table being distributed among the data-storage facilities;

- each of the tables having an associated Star Map, S1 through SN, respectively, each Star Map being distributed among the data-storage facilities,

- each Star Map including bitmap entries having locations indexed by ~~the~~ a hash of one or more values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates ~~the presenece of a row~~ is present in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry;

- a join process executed on one or more of the plurality of CPUs that cause the CPUs to

- a) logically AND the Star Maps S1 through SN to produce Star Map SJ;

b) for each set bitmap entry in SJ, search T1 through TN for all rows having values in their one or more join key columns that hash to the location of the set bitmap entry; and

c) join the resulting rows to produce a join result.

32. (Canceled)

33. (Original) The database system of claim 31 where when logically ANDing the Star Maps the process

sets an entry in SJ having a location defined by a hash value if the entries in all of S1 through SN having locations defined by the hash value are set.

34. (Original) The database system of claim 31 where when logically ANDing the Star Maps the process

a1) performs a logical AND of S1 and S2 to produce SJ;

a2) sets i to 3;

a3) while i is less than or equal to N, performs a4) through a6);

a4) performs a logical AND of Si with SJ to form Stemp;

a5) sets SJ equal to Stemp;

a6) increments i.

35. (Original) The database system of claim 34 where S1 through SN and SJ are the same size and when performing a logical AND of Si with SJ to form Stemp the process

ANDs the entry at a location in Si defined by a hash value with the entry at a location in SJ defined by the hash value.

36. (Original) The database system of claim 31 where each of T1 through TN are indexed by the hash of the combined entries in its respective one or more join key columns, and when searching T1 through TN for rows having values in its one or more columns that hash to the location of the set bitmap entry the process

uses the hash value associated with the location of the set bitmap entry as an index to retrieve rows from tables T1 through TN.

37. (Original) The database system of claim 31 where one or more of the Star Maps is a table and a first portion of the hash value that indexes the locations of a Star Map defines a row within the Star Map and a second portion of the hash value defines an offset within the defined row.

38. (Original) The database system of claim 37 where the first portion is the first half of the hash value and the second portion is the second half of the hash value.

39. (Original) The database system of claim 37 where the hash value is 32 bits long, the first portion is the first 16 bits of the hash value, and the second portion is the second 16 bits of the hash value.

40. (Original) The database system of claim 31 where
at least one of tables T1 through TN is ordered by the values of its one or more join key columns.

41. (Original) The database system of claim 31 where each entry in a Star Map is one bit.

42. (Original) The database system of claim 31 where each entry in a Star Map is sixteen bits.

43. (Original) The database system of claim 31 where each Star Map entry includes one or more bits and each bit corresponds to the hash of one or more values associated with the one or more join key columns of its associated table.

44. (Currently amended) A data structure stored in a memory for use in performing a join in a database management system executed by a computer, the data structure comprising a Star Map associated with a table referenced in a query, wherein the Star

Map includes bitmap entries having locations indexed by a combination of one or more values associated with one or more join key columns of its associated table, where a bitmap entry in the Star Map, if set, indicates ~~the presence of a row~~ is present in the associated table that has entries in the one or more join key columns that, when combined, identify the location of the bitmap entry.

45. (Currently amended) A method for joining a plurality of tables T1 through TN, where each of the tables has an associated Star Map, S1 through SN, respectively, and each Star Map includes bitmap entries having locations indexed by one or more values associated with a join key column of its associated table, where a bitmap entry in a Star Map, if set, indicates ~~the presence of a row~~ is present in its associated table that has an entry in the join key column that identifies the location of the bitmap entry, the method including:

- a) performing a Boolean operation using the bitmap entries of the Star Maps S1 ~~through SN to produce a join Star Map SJ;~~
- b) using SJ to select rows from the tables T1 through TN; and
- c) joining the resulting rows to produce a join result.

46. (Original) The method of claim 45 where using SJ to select rows from the tables T1 through TN includes for each set bitmap entry in SJ, searching T1 through TN for all rows having values in their respective join key columns that identify the location of the set bitmap entry.

47. (Original) The method of claim 46 where

each bitmap entry in each Star Map has an associated row number and column number;

the row number and the column number for a set bitmap entry in a Star Map can be combined to produce a value that corresponds to the value of the join key column of the Star Map's associated table; and

at least one of the tables T1 through TN is ordered by the values of its join key column.

48. (New) A method for joining a plurality of tables T1 through TN to produce a join result, where each of the tables has an associated Star Map, S1 through SN, respectively, and each Star Map includes bitmap entries having locations indexed by a hash of one or more values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates a row is present in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry, the method including:

- predicting cardinality of the join result;

- determining if the predicted cardinality is less than or equal to a predefined threshold value; and

- if the predicted cardinality is less than or equal to a predefined threshold value,

- logically ANDing the Star Maps S1 through SN to produce Star Map SJ,

- for each set bitmap entry in SJ, searching T1 through TN for all rows having values in their one or more join key columns that hash to the location of the set bitmap entry, and

- joining the resulting rows to produce the join result.

49. (New) A computer program, stored on a tangible storage medium, for use in joining a plurality of tables T1 through TN to produce a join result, where each of the tables has an associated Star Map, S1 through SN, respectively, and each Star Map includes bitmap entries having locations indexed by a hash of one or more values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates a row is present in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry, the program including executable instructions that cause a computer to

- predict cardinality of the join result;

- determine if the predicted cardinality is less than or equal to a predefined threshold value; and

- if the predicted cardinality is less than or equal to a predefined threshold value

- logically AND the Star Maps S1 through SN to produce Star Map SJ,

for each set bitmap entry in SJ, search T1 through TN for all rows having values in their one or more join key columns that hash to the location of the set bitmap entry, and

join the resulting rows to produce the join result.

50. (New) A database system for accessing a database, the database system including a massively parallel processing system including

one or more nodes;

a plurality of CPUs, each of the one or more nodes providing access to one or more CPUs;

a plurality of virtual processes each of the one or more CPUs providing access to one or more processes;

each process configured to manage data stored in one of a plurality of data-storage facilities;

a plurality of tables T1 through TN, each table being distributed among the data-storage facilities;

each of the tables having an associated Star Map, S1 through SN, respectively, each Star Map being distributed among the data-storage facilities,

each Star Map including bitmap entries having locations indexed by a hash of one or more values associated with one or more join key columns of its associated table, where a bitmap entry in a Star Map, if set, indicates a row is present in its associated table that has entries in the one or more join key columns that together hash to the location of the bitmap entry;

a join process executed on one or more of the plurality of CPUs to produce a join result that cause the CPUs to

predict cardinality of the join result;

determine if the predicted cardinality is less than or equal to a predefined threshold value; and

if the predicted cardinality is less than or equal to a predefined threshold value logically AND the Star Maps S1 through SN to produce Star Map SJ,

for each set bitmap entry in SJ, search T1 through TN for all rows having values in their one or more join key columns that hash to the location of the set bitmap entry, and
join the resulting rows to produce the join result.